AF/2700
#8
6/28/02 A.W.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

MARK A. BAKKE et al.

Serial No.: 09/373,795

Filed: August 13, 1999

For: DISTRIBUTED FILE DATA LOCATION

Attorney Docket No.: 98-127-NSC/STK98127PUS

Group Art Unit: 2177

Examiner: S. Channavajjala

APPEAL BRIEF

Box AF
Commissioner for Patents
United States Patent and Trademark Office
Washington, D.C. 20231

Sir:

This is an appeal brief from the final rejection of claims 1-20 of the Office Action dated February 8, 2002. This application was filed on August 13, 1999.

I. REAL PARTY IN INTEREST

The real party in interest is Storage Technology Corporation, a corporation organized and existing under the laws of the state of Delaware, and having a place of business a One StorageTek Drive, Louisville, Colorado, 80028-4309, as set forth in the assignment recorded in the U.S. Patent and Trademark Office on August 13, 1999, at Reel 010173/Frame 0284.

CERTIFICATE OF MAILING UNDER 37 C.F.R. § 1.8

I hereby certify that this paper, including all enclosures referred to herein, is being deposited with the United States Postal Service as first-class mail, postage pre-paid, in an envelope addressed to: Box AF, Commissioner for Patents, United States Patent and Trademark Office, Washington, D.C. 20231 on:

June 17, 2002	Mark D. Chuey, Ph.D.	
Date of Deposit	Name of Person Signing	Signature

II. RELATED APPEALS AND INTERFERENCES

There are no appeals or interferences related to the present appeal.

III. STATUS OF CLAIMS

Claims 1-20 are pending in this application. Claims1-20 have been rejected and are the subject of this appeal.

IV. STATUS OF AMENDMENTS

No amendment after final rejection was filed. No amendments are pending in this application.

V. SUMMARY OF THE INVENTION

With reference to Figure 2, a file system 50 for storing data is provided. File system 50 includes a plurality of storage devices 54, each storage device 54 operative to store at least one copy of at least one file. At least one location server 66 maps a file identifier for each file into the location of each copy of the file represented by the file identifier. At least one name server 60 maps a file name to the file identifier referenced by the file name. Each name server 60 is physically separate from each location server 66.

VI. ISSUES

The following art, cited in the Examiner's rejections of claims 1-20, is referenced in this brief: U.S. Patent No. 6,081,807 to Story *et al.* (Story) and U.S. Patent No. 6,029,168 to Frey (Frey).

1. Whether claim 1 is unpatentable under 35 U.S.C. § 103(a) over Story in view of Frey.

2. Whether claims 16 is unpatentable under 35 U.S.C. § 103(a) over Story in view of Frey.

3. Whether claim 6 is unpatentable under 35 U.S.C. § 103(a) over Story in view of Frey.

4. Whether claim 11 is anticipated under 35 U.S.C. § 102(e) by Story.

5. Whether Story teaches or suggests limitations in claims 2, 12 and 17.

6. Whether Story teaches or suggests limitations in claims 3, 13 and 18.

VII. GROUPING OF CLAIMS

A. Independent claim 1 and claims 4, 5 and 7-10 depending therefrom are patentable.

B. Independent claim 16 and claims 19 and 20 depending therefrom are patentable. Claim 16 merits separate consideration in that claim 16 provides for a client not found in claim 1.

C. Claim 6 depends from claim 1. Claim 6 merits separate consideration in that claim 6 provides for name servers operating under different file access standards.

D. Independent claim 11 and claims 14 and 15 depending therefrom are patentable. The Examiner rejected claim 11 under § 102. Claim 11 therefore merits separate consideration from independent claims 1 and 16, which were rejected under § 103.

E. Dependent claims 2, 12 and 17 merit separate consideration since each provides for files saved as file extents.

F. Dependent claims 3, 13 and 18 merit separate consideration since each provides for files represented in storage as objects with each file identifier an object identifier.

VIII. ARGUMENT

In a final Office Action dated February 8, 2002, the Examiner rejected claims 1-20 in the above-captioned patent application. Appellants respectfully disagree with these rejections based on the following arguments.

1. Whether claim 1 is unpatentable under 35 U.S.C. § 103(a) over Story in view of Frey.

The Examiner rejected claim 1 under 35 U.S.C. § 103(a) over Story in view of Frey. Claim 1 provides for a file system for storing data. A plurality of storage devices store at least one copy of at least one file. At least one location server maps a file identifier for each file into the location of each copy of the file represented by the file identifier. At least one name server maps a file name to the file identifier referenced by the file name. *Each name server is physically separate from the at least one location server.* No combination of Story and Frey teach or suggest Appellants' claim 1.

As seen in Frey's Figure 5, whatever mapping Frey discloses is accomplished in data node 42. Thus, Frey does not disclose separate servers for mapping names and file identifiers.

The Examiner identified Story's name server 130 as Appellants' name server. (Final Office Action, page 4.) Story's disclosure for name server 130 is provided in column 5, lines 7-8, as follows:

> [N]ame server 130 is responsible for file name hierarchy and provides pathname resolution.

The Examiner identified Story's NFS server 122 as Appellants' location server. (Final Office Action, page 4.) However, both NFS server 122 and name server 130 reside in Story's network server 106. Story's Figure 1 shows NFS server 122 and name server 130 within network server 106. This arrangement is described in column 4, lines 4-20, which is reproduced as follows (emphasis added):

> *In network server 106, NFS server 122* is linked, via an interface 126, to a disk file system, such as an OSS (Open Systems Services) file system 124 developed by and available from Tandem Computers Incorporated, Cupertino, Calif. The phrase "file system" has several distinct meanings in computer science. As used in the application, unless otherwise qualified, a "file system" refers to a body of software designed to store, organize, protect, and retrieve data using some storage medium such as

> disk. OSS file system 124 is linked to a disk process 128 and a *name server 130* which is also linked to disk process 128. Disk process 128 is linked to a data storage device 132 and a file manager 134. It should be understood that, in the embodiment of FIG. 1, *elements* 112, 114, 116, 118, *122*, 124, 126, 128, *130* and 134 *are implemented as software programs stored in memory and executed by one or more respective processors* (not shown).

Story's NFS server 122 and name server 130 are software modules running on the same network server 106. Thus, neither Story nor Frey teach or suggest Appellants' physically separate name server and location server.

Claims 4, 5 and 7-10 depend from claim 1 and are therefore also patentable.

2. Whether claim 16 is unpatentable under 35 U.S.C. § 103(a) over Story in view of Frey.

Independent claim 16 provides a file system for storing data. The file system includes a plurality of storage devices, at least one location database, at least one name database and at least one client. The location database has a map between a file identifier for each file and location information for each copy of the file represented by the file identifier. The name database has a map between a file name and the file identifier referenced by the file name. Each name database is physically separate from any location database. Each client requests a file identifier corresponding to a requested file name, receives the file identifier mapped to the requested file name, requests location information corresponding to the received file identifier, receives location information mapped to the received file identifier, and accesses data using the location information.

a. Story doe not teach or suggest <u>Appellants' name database</u>

Claim 16 provides for at least one name database operating with file names and file identifiers referenced by the file name. The Examiner identified Story's name server 130

as Appellants' name database. Story's *entire* disclosure for name server 130 is provided in column 5, lines 7-8, as follows:

> [N]ame server 130 is responsible for file name hierarchy and provides pathname resolution.

Thus, Story neither teaches nor suggests a name database that receives a name and returns a file identifier corresponding to that name.

In response to this argument, the Examiner replied with the following at page 22:

> Story teaches name database [col 4, line 12-14, fig 1, element 130], examiner interpreting name database corresponds to Story's fig 1, element 130, further name server or database is responsible for organizing file names in hierarchical structure and returns file identifier corresponding to that name as detailed in col 5, line 7-8.

The Examiner's reply does not address Appellants' argument. The Examiner has not located any teaching in Story of name database 130 returning a file identifier corresponding to a received file name.

b. Story does not teach or suggest ***Appellants' location database***

Claim 16 provides for at least one location database operating with a file identifier for each file and the location of each copy of the file represented by the file identifier. The Examiner identified Story's NFS server 122 as Appellants' location database. Story discloses that NFS server 122 receives a file handle including a file set ID and a file ID in column 4, lines 41-47. Story also discloses that the NFS server forwards the file set ID and file ID to interface 126 at column 5, lines 2-6. Thus, if the file handle is the file identifier, Story's NFS server 122 receives and sends a file identifier and does not send file location information.

In reply to this argument, the Examiner provided the following at page 22:

> Story teaches for example location database, see fig 1, element 122], examiner interpreting location database corresponds to Story's fig 1, element 122, NFS server.

The Examiner can find no teaching of Appellants' location database. Story's NFS server does not provide a map between a file identifier for each file and location information for each copy of the file represented by the file identifier as provided by claim 16.

c. ***Neither Story nor Frey teach or suggest separate name database and location database***

Claim 16 provides that each name database be physically separate from any location database. As seen in Frey's Figure 5, whatever mapping Frey discloses is accomplished in data node 42. Further, the Examiner identified Story's name server 130 as Appellants' name database and identified Story's NFS server 122 as Appellants' location database, both of which reside in Story's network server 106. (*See*, col. 4, ll. 4-14.) Thus, neither Story nor Frey teach or suggest Appellants' physically separate name database and location database.

d. ***Neither Story nor Frey teach or suggest Appellants' client***

Claim 16 provides for a client operative to request a file identifier corresponding to a requested file name, receive the file identifier mapped by a name database to the requested file name, request location information corresponding to the received file identifier, receive location information mapped by a location database to the received file identifier, and access data using the location information. The Examiner asserts that Story's NFS client 116 in network client 102 is Appellants' client. However, Story discloses receiving data and/or status as a direct result of sending a request. (*See*, Story's Figure 2.) No location information passes to NFS client 116.

Claims 19 and 20 depend from claim 16 and are therefore also patentable.

3. Whether claim 6 is unpatentable under 35 U.S.C. § 103(a) over Story in view of Frey.

Claim 6, which depends from claim 1, provides that the at least one name server is a plurality of name servers. At least one of the plurality of name servers operates under a first file access standard and at least one of the plurality of the name servers operates under a second file access standard different from the first file access standard.

The Examiner indicates NFS as a first standard in Story. The Examiner asserts that a second file access standard is disclosed in Story at column 6, line 64, through column 7, line 12, as follows:

> The file system obtains the information about the current access state of the file from an associated VNODE. The current access state may be read-only, read-and-write, or closed.
>
> At step 504, the file system determines whether the current state of the file matches the needed state indicated by the read or write request. If there is a match, then no further processing is needed. The file already has the appropriate state for the requested operation. If the current file state is not the same as the needed state, the file system then determines whether the current file state is a read and write state at step 508. If so, the file again has the appropriate state for the requested operation and the processing is completed. However, if the current file state is not a read and write state, the file system determines whether the current file state is closed at step 510.

This passage has nothing whatsoever to do with different file access standards. Further, in neither Office Action has the Examiner identified anything in Story as Appellants' second file standard. Story neither teaches nor suggests the use of a second standard, let alone the ability for name servers to operate under different file access standards.

4. Whether claim 11 is anticipated under 35 U.S.C. § 102(e) by Story.

Independent claim 11 provides for a method for accessing a file referenced by a file name. The file name is sent to a name server. A file identifier corresponding to the file name is received from the name server. The file identifier is sent to a location server which

is separate from the name server. File location information corresponding to the file identifier is received from the location server. The file is accessed using the location information.

Story discloses interfacing with a stateless NFS server. As illustrated in Story's Figure 1, network client 102 communicates with network server 106 using network 110. Network client accesses local disk storage 120 and network server 106 accesses network storage 132. As such, Story discloses a system similar to the prior art system illustrated in Appellants' Figure 1 and described by Appellants from page 5, line 8, through page 6, line 2, reproduced as follows (emphasis added):

> Referring to Figure 1, a schematic diagram illustrating a prior art client-server relationship is shown. A file system, shown generally by 20, includes clients 22 accessing data held in files on one or more storage devices 24. Each storage device 24 is accessed through a server, one of which is indicated by 26. A typical example of file system 20 is the Unix-based *Network File System (NFS)*. In NFS, client 22 wishing to access data first *forwards the name of the file containing the data to server 26*, as shown by 28. Server 26 returns a handle to the requested file as indicated by 30. Client 22 then *forwards a data request with the received handle to server 26*, as indicated by 32. Server 26 requests the data from storage device 24, as indicated by 34. Storage device 24 returns the data to server 26, as indicated by 36, and server 26 forwards the data to client 22, as indicated by 38. Another typical example of file system 20 is embodied in the CIFS (Common Internet File System) found in the WINDOWS NT ® server by Microsoft Corp. Server 26 provides a file descriptor in response to a name provided by client 22. Client 22 uses the file descriptor to access data through a logical connection through server 26 to storage device 24. The file descriptor is valid only for the life of the connection.
>
> There are several problems associated with the traditional client-server system. First, server 26 may not have sufficient resources to support an increasing number of clients 22. Second, the failure of server 26 makes storage device 24 inaccessible by clients 22. Third, a client 22 not directly connected to server 26 may have difficulty locating and accessing a file stored on storage device 24 connected to server 26. Finally, server 26 may not be able to properly respond to

> client 22 requesting a file using a naming scheme different than the scheme used by server 26.

Story fails to anticipate Appellants' claim 11 for at least the following four reasons.

a. Story does not disclose separate ***name server and location server***

The Examiner has identified Story's name server 130 as Appellants' name server. The Examiner has also identified Story's NFS server 122 as Appellants' location server. Claim 11 provides that these servers are separate. However, Story discloses name server 130 and NFS server 122 both in network server 106. (*See*, col. 4, ll. 4-14.) Thus, Story neither teaches nor suggests separate name and location servers.

In reply to this argument, the Examiner provided the following at page 20:

> Examiner disagree with the applicant because, firstly Story teaches for example network file system or NFS [see fig 1], secondly, Story's teaching including i) file system element 114, name server element 130 as detailed in fig 1, thirdly, network client element 102 is connected through network element 110 to network server element 106, see fig 1.

This somewhat cryptic statement seems to infer that Story somehow teaches separate name and location servers because network client 102 is separated from network server 106 by network 110. However, as was pointed out to the Examiner, the two elements identified by the Examiner as Appellants' location server and name server, namely NFS server 122 and name server 130, respectively, *are both located in Story's network server 106*. Thus, the Examine's argument concerning Story's client element 102 and network element 110 is irrelevant.

b. Story does not disclose a name server receiving a file name and sending a file identifier ***corresponding to the file name***

Claim 11 provides for sending a file name to the name server and receiving a file identifier corresponding to the file name from the name server. The Examiner identified

Story's name server 130 as Appellants' name server. Story's disclosure for name server 130 is provided in column 5, lines 7-8, as follows:

> [N]ame server 130 is responsible for file name hierarchy and provides pathname resolution.

Thus, Story neither teaches nor suggests a name server that receives a name and returns a file identifier corresponding to that name.

In reply to this argument, the Examiner provided the following at page 20:

> Examiner disagree with the applicant because firstly, name server is responsible for file name hierarch and provides pathname, secondly, Story teaches for example file handle which contains information such as type of file, unique identifier or file ID as detailed in col 4, line 41-47.

The Examiner is mixing up his own construction of Appellants' invention. The passage cited by the Examiner refers to Story's NFS server 122, which the Examiner has identified as Appellants' location server, not Appellants' name server. The passage is reproduced as follows:

> The NFS LAN interface process dispatches work to a server process in NFS server 122, based on the contents of a "file handle," which is a parameter in the NFS request. A file handle contains such information as the type of file, the time of creation of the file, a unique identifier (file set ID) for the file set in which the file resides, a unique identifier (file ID) for the file within the file set, etc.

The interface process is in network server 106 receives requests from NFS client 116 in network client 102. (*See*, col. 4, lines 31-37.) Thus, the Examiner has provided no basis for his assertion that Story's name server 130 receives a file name and returns a file identifier corresponding to the file name.

c. ***Story does not disclose a location server receiving a file identifier and sending file location information***

Claim 11 provides for sending the file identifier to a location server and receiving file location information corresponding to the file identifier from the location server. The file is then accessed using the file location information. The Examiner identified Story's NFS server 122 as Appellants' location server. Story discloses that NFS server 122 receives a file handle including a file set ID and a file ID in column 4, lines 41-47. Story also discloses that the NFS server forwards the file set ID and file ID to interface 126 at column 5, lines 2-6. Thus, if the file handle is the file identifier, Story's NFS server 122 receives and sends a file identifier and does not send file location information.

The Examiner replied to this argument with the following at page 20:

> Applicant agrees in the response that Story discloses NFS server containing file set ID and file ID interfaced through element 126 as detailed in col 5, line 2-6, further Story's NFS server 122 is capable of both sending and receiving file information such as file identifier, location information and like because they are connected through network element 110 as detailed in fig 1.

The Examiner asserts that Story discloses file location information sent from Story's NFS server. The cited passage is reproduced as follows:

> OSS file system 124 includes a hashing mechanism for locating a VNODE associated with a file based on the file ID and file set ID in the file handle that is included in the NFS client request.

A VNODE is defined in Story at column 4, lines 56-62, as follows:

> A VNODE contains various kinds of information about the state of the file, including whether it is open, where its cached data (if any) is located, the timestamps associated with the file, etc. A VNODE also contains an NFS "pseudo-open" status of the associated file. A pseudo-open describes the state of a file currently being accessed via the NFS server.

Thus, a VNODE does not indicate the location of a file[1]. Further, there is no disclosure in Story that whatever information is held in a VNODE is sent from NFS server 122 in response to a received file identifier as provided in claim 11.

d. *Story does not disclose a file identifier received from a name server and sent to a location server*

Claim 11 provides for receiving a file identifier corresponding to the file name from the name server and sending this file identifier to the location server. Thus, whatever the Examiner identifies as the file identifier must come from whatever the Examiner identifies as the name server and be sent to whatever the Examiner identifies as the location server. The Examiner identified Story's file ID as Appellants' file identifier. There is no disclosure in Story of any file ID that is received from Story's name server 130 and is sent to Story's NFS server 122.

In reply to this argument, the Examiner provided the following at pages 20-21:

> Examiner disagree with the applicant Story specifically teaches for example file identifier [see col 4, line 43-47, fig 4, element 404], more specifically file handler contains contains file identifier information, and all the requests would be processed through NFS server element 122 as detailed in col 4, line 44-52].

Whether this is true or not, the Examiner still has not found any teaching of Appellants' invention as provided in claim 11. The Examiner must find some disclosure of a file identifier received from the name server and sent to the location server. If the Examiner maintains his assertion as to Story's anticipation of Appellants' invention, the Examiner must find a file identifier that is received from Story's name server 130 and is sent to Story's NFS server 122.

Claims 14 and 15 depend from claim 11 and are therefore also patentable.

---

[1]One of ordinary skill in the art of file systems would not consider the location of cached data to be the location of the file from which the cached data was obtained.

5. Whether Story teaches or suggests limitations in claims 2, 12 and 17.

Claim 2 depends from claim 1. Claim 12 depends from claim11. Claim 17 depends from claim 16. Each of claims 2, 12 and 17 provide that a file is stored as at least one file extent and that the file identifier includes a file handle. The Examiner rejected claims 2 and 17 under 35 U.S.C. § 103(a) over Story in view of Frey. The Examiner rejected claim 12 under 35 U.S.C.§ 102(e) as anticipated by Story. Neither Story nor Frey mention an extant. Further, the Examiner made no argument for a teaching or suggestion of an extent in any reference. Appellants pointed out this defect in the Examiner's arguments in a response following the first Office Action. The Examiner chose to repeat, verbatim, his argument in the final Office Action rather than to identify any extent in Story or anywhere else.

6. Whether Story teaches or suggests limitations in claims 3, 13 and 18.

Claim 3 depends from claim 1. Claim 13 depends from claim11. Claim 18 depends from claim 16. Each of claims 3, 13 and 18 provide that a file is represented in storage as an object and that each file identifier is an object identifier. The Examiner rejected claims 3 and 18 under 35 U.S.C. § 103(a) over Story in view of Frey. The Examiner rejected claim 13 under 35 U.S.C.§ 102(e) as anticipated by Story. Neither Story nor Frey teach or suggest Appellants' representation of files as objects and use of an object identifier as a file identifier.

The Examiner rejected claim 13 with the following argument at page 13:

> As to Claim 13, Story details a system which including 'each file is represented in storage as an object and each file identifier is an object identifier' [col 4 lin 53-60, col. 5, line 2-6], examiner interpreting object identifier corresponds to Story's VNODE because, OSS file system has the ability to locate VNODE associated with the file using hashing mechanism based on the file ID and the file including file handle as detailed in col 5 line 30-34.

Claims 3 and 18 were rejected using similar language. The sections in Story cited by the Examiner are reproduced as follows:

> OSS file system 124 supports disk files. It contains one or more file-system data structures called VNODEs (virtual nodes). Each file currently in use in the server has an associated VNODE. A VNODE contains various kinds of information about the state of the file, including whether it is open, where its cached data (if any) is located, the timestamps associated with the file, etc. A VNODE also contains an NFS "pseudo-open" status of the associated file.
>
> * * * *
>
> OSS file system 124 includes a hashing mechanism for locating a VNODE associated with a file based on the file ID and file set ID in the file handle that is included in the NFS client request.
>
> * * * *
>
> Upon receiving the READ or WRITE NFS request, the OSS file system then attempts to locate a VNODE associated with the file using a hashing mechanism in the OSS file system at step d, based on the file ID and file set ID included in the file handle. If no VNODE is found, one is created.

The passages cited by the Examiner appear to have nothing whatsoever to do with either objects or object identifiers. Further, the Examiner's only connection between Story and Appellants' objects is that Story's "OSS file system has the ability to locate VNODE associated with the file using hashing mechanism based on the file ID." (Final Office Action, pg. 13.) Hashing may be defined as follows:

> **hashing**: A method of transforming a search key into an address for the purpose of storing and retrieving items of data. The method is often designed to minimize the search time.[2]

Hashing has nothing whatsoever to do with objects. The two are independent and distinct concepts. The Examiner has failed to find any teaching or suggestion of Appellants' representation of files as objects and use of an object identifier as a file identifier.

---

[2]IBM Dictionary of Computing, 10^{th} Ed., McGraw-Hill, Inc., 1994, pg. 309.

IX. CONCLUSION

Appellants believe that claims 1-20, all claims pending in the application under appeal, are patentable. Appellants respectfully request that the Board so hold.

The fee of $320 as applicable under the provisions of 37 C.F.R. § 1.17(c) is enclosed. Please charge any additional fee or credit any overpayment in connection with this filing to our Deposit Account No. 02-3978. A duplicate of this notice is enclosed for this purpose.

Respectfully submitted,

MARK A. BAKKE et al.

By: ________________

Mark D. Chuey, Ph.D.
Registration No. 42,415
Attorney/Agent for Appellant

Date: June 17, 2002

BROOKS & KUSHMAN P.C.
1000 Town Center, 22nd Floor
Southfield, MI 48075
Phone: 248-358-4400
Fax: 248-358-3351

Enclosure - Appendix

X. APPENDIX - CLAIMS ON APPEAL

1. A file system for storing data comprising:

a plurality of storage devices, each storage device operative to store at least one copy of at least one file;

at least one location server operative to map a file identifier for each file into the location of each copy of the file represented by the file identifier; and

at least one name server operative to map a file name to the file identifier referenced by the file name, each name server physically separate from the at least one location server.

2. A file system as in claim 1 wherein each file is stored as at least one file extent, the file identifier comprising a file handle.

3. A file system as in claim 1 wherein each file is represented in storage as an object and each file identifier is an object identifier.

4. A file system as in claim 1 wherein each location server is further operative to store metadata associated with each file identifier.

5. A file system as in claim 1 wherein at least one location server is on a first computer system and at least one name server is on a second computer system in communication with the first computer system.

6. A file system as in claim 1 wherein the at least one name server is a plurality of name servers, at least one of the plurality of name servers operating under a first file access standard and at least one of the plurality of the name servers operating under a second file access standard different from the first file access standard.

7. A file system as in claim 1 further comprising at least one client, each client operative to:

request a file identifier for a new file from one of the at least one location server;

receive the requested file identifier;

register the file identifier and a new file name for the new file with at least one name server.

8. A file system as in claim 7 wherein each client is further operative to:

send a requested file name to the name server;

receive a file identifier corresponding the requested file name and an indicated location server from the name server;

request from the indicated location server updated locations for a write operation to the requested file;

receive updated locations from the location server; and

write data to the received updated locations.

9. A file system as in claim 7 wherein each client is further operative to:

send a requested file name to the name server;

receive a file identifier corresponding the requested file name and an indicated location server from the name server;

request from the indicated location server the location of data corresponding to the file identifier;

receive at least one requested location; and

read data from the at least one received requested location.

10. A file system as in claim 7 wherein each client is further operative to:

send an existing file name for an existing file to the name server;

receive a file identifier corresponding the existing file from the name server;

send the file identifier and a new name for the existing file to at least one name server, thereby registering the new file name for the existing file.

11. A method for accessing a file referenced by a file name, the file stored on at least one storage device, the method comprising:

sending the file name to a name server;

receiving a file identifier corresponding to the file name from the name server;

sending the file identifier to a location server, the location server separate from the name server;

receiving file location information corresponding to the file identifier from the location server; and

accessing the file using the location information.

12. A method for accessing a file as in claim 11 wherein each file is stored as at least one file extent, the file identifier comprising a file handle.

13. A method for accessing a file as in claim 11 wherein each file is represented in storage as an object and each file identifier is an object identifier.

14. A method for accessing a file as in claim 11 further comprising accessing file metadata stored in the location server.

15. A method for accessing a file as in claim 11 further comprising sending the file identifier and a new file name to at least one name server, thereby registering the new name for the file.

16. A file system for storing data comprising:

a plurality of storage devices, each storage device operative to store at least one copy of at least one file;

at least one location database comprising a map between a file identifier for each file and location information for each copy of the file represented by the file identifier;

at least one name database comprising a map between a file name and the file identifier referenced by the file name, each name database physically separate from the at least one location database; and

at least one client operative to

(a) request a file identifier corresponding to a requested file name,

(b) receive the file identifier mapped to the requested file name,

(c) request location information corresponding to the received file identifier,

(d) receive location information mapped to the received file identifier, and

(e) access data using the location information.

17. A file system as in claim 16 wherein each file is stored as at least one file extent, the file identifier comprising a file handle.

18. A file system as in claim 16 wherein each file is represented in storage as an object and each file identifier is an object identifier.

19. A file system as in claim 16 wherein the client is further operative to access file metadata stored in the location database.

20. A file system as in claim 16 wherein the client is further operative to send the file identifier and a new file name to at least one name database, thereby registering the new name for the file.